

Denormalization Decisions:

Our local/domestic table is empty, only receiving the flightId from schedule, but this is because only local/domestic flights offer the extra charges class. Although they could have been placed under the table as an attribute, it would have resulted in multi-valued attributes which is bad design. Furthermore, having the local/domestic class represented allows those types of flights to be easier to identify, and thereby simplifying the queries.

Besides that, we did not make any denormalization decisions, because doing so would introduce a massive amount of redundancy to our database. Had we denormalized a few classes, such as crew and flightSchedule, it would have propagated irrelevant information onto other classes as well. For example, flightId would have to be assigned to crew members and that would have resulted in our queries returning multiples of the same crew members. Also, the data we have in our database isn't large enough to consider any saved time that comes from loading denormalized data. Even with all the joins in the queries, the time saved would not make any noticeable difference. Data insertions would also become a problem. They would require more columns to be entered and that would result in time being lost compared to making other denormalization decisions.